# Design and Implementation of an Energy Efficient, Parallel, Asynchronous DSP

W.Suntiamorntut[1], L.E.M.Brackenbury[2] and Jim Garside[2]
[1] Department of Computer Engineering,
Prince of Songkla University
Hatyai, Songkhla, 90112, Thailand
[2] School of Computer Science, University of Manchester
Manchester, M13 9PL, UK
E-mail: wannarat@coe.psu.ac.th, {lbrackenbury, jdg}@cs.man.ac.uk

**Abstract:** Energy efficient computing in a DSP has become an important research issue in order to have a longer battery operating time to support the modern portable devices. The energy efficient functional unit has been designed and implemented for an in-house asynchornous DSP named, Configurable Asynchronous DSP for Reduced Energy (CADRE). CADRE-successor (CADRE-s) has been implemented as a full custom design and simulations are presented to successfully demonstrate the energy-efficiency of the FU. The results show that the FU designed can achieve an energy improvement by a factor of 5 in the multiply accumulator units and a factor of nearly 2 for the overall system compared with the original CADRE system. This demonstrates the importance that energy efficient logic, circuit and layout techniques contribute to a design.

## 1. Introduction

An asynchronous parallel architecture, named CADRE (Configurable Asynchronous DSP for Reduced Energy)[1] was designed in the School of Computer Science, University of Manchester and expected to use in portable applications. CADRE expanded instructions to give a flexible VLIW capability including a large register file, instruction buffer and four functional units. It has been design based on the sign magnitude number representation and asynchronous circuit design. Even though CADRE had an efficient architecture for DSP processors, it required a large amount of power as demonstrated in[1,2].

CADRE was implemented by exploiting four-way parallelism, as this appeared to be optimal for power reduction[3]. This was based on the premise area can be traded for increased speed because silicon area is rapidly becoming less expensive. As well-known, arithmetic operations, such as multiply, add and multiply-accumulate are frequently performed and consumed hungry power. From the previous work using random plus speed data, a large percentage of power was found to be dissipated in the Multiply Accumulator Units (MAC units) amounting to about 50% of the overall power consumption.

Therefore, the functional unit (FU) has been re-designed and re-implemented to demonstrate the energy saving improvement possible. To reduce the power consumption in the new FU, the major dominant components, the multiplier and adder, are designed and implemented with coherent low power techniques. The new four-way parallel asynchronous DSP which has been designed, implemented and tested is called CADRE-s (CADRE successor) and will be referred to as CADRE-s throughout in this paper.

This paper is organized as follows. In Section II we describe the CADRE-s Top-Level Architecture. Coherent low power techniques using in the new FU is illustrated in Section III. The implementation is shown in Section IV. Finally, Section V concludes this paper.

## 2. CADRE-s Top-level Architecture

The top-level architecture for CADRE-s has been designed to demonstrate the energy efficiency of the functional unit and the other advantageous features of CADRE-s such as four-way parallelism and configurable memories. In this top-level architecture, the 32x64 bit configuration memory has been attached to each FU and it stores the opcode. In contrast with the original CADRE, the operands of each FU in the current architecture are fetched directly from on-chip RAMs (OPA and OPB). The new system consists of four FUs connected together with a global bus named the Global Interface Functional Unit (GIFU), whilst each pair of FUs are connected locally via the bus named Local Interface Functional Unit (LIFU). The output data from each FU is stored in another on-chip RAM. Two RAMs are used for the top level control. One is the top-level 14-bit instruction and is stored in a program memory. The second one is the address RAM which effectively performs the function of a 16-bit program counter (PC).

CADRE-s employs a scan path structure for downloading instructions/data and uploading results as shown in Figure1. There are five separate serial scan paths in the design, one is for the address and program memories and the others used for the operand, configuration and result memories. This makes the system fully testable as internal states can be controlled and monitored.

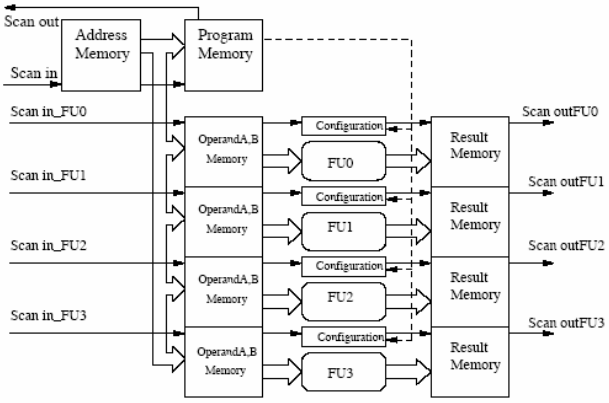


Figure1 CADRE-s Data Organization

### 2. 1 MIMD with VLIW encoding

In CADRE-s, each FU has its own operands, configurable and result memories. Hence, each FU can

execute independently. It differs from the FU that has been used in an asynchronous superscalar[4]. It has shared memories and registers. A special mechanism for the instruction dispatch unit is therefore required. Because CADRE-s contains four FUs which work independently, the instruction is relatively long. To minimize the length, the instruction has been divided into two parts. One part is stored in the configuration memory and the other part in the top level instruction memory. The configuration memory contains the FU opcode, input/output selection, and the shift condition whilst the top level instruction contains four enable signals plus four accumulator write enable control signals and a common 6-bit address for the configuration memory. Storing the 6-bit address of configuration memory not only reduces the number of instruction bits but the instruction can be compressed. This means that the instructions can be recalled when the same instructions but different data are required. Most DSP algorithms can take advantage of this feature to reduce the size of the program.

### 2. 2 Five Stage Asynchronous Pipeline

The CADRE-s pipelined processing unit is organized using self-timing techniques. An asynchronous pipeline operates at a variable rate determined by current conditions, unlike a single clock system, where the whole pipeline will be clocked at a rate determined by the worst-case delay in the slowest stage. Although, a multi-clocked system has been proposed in [5-6] to allow a variable rate operation, this method is limited by the number of the various clock cycles which are generated by the clock generator. In a clockless system, the next instruction can start as soon as the previous result is generated. Therefore, the self-timed system is able to operate at the average-case performance rather than worst-case.

## 3. Low Power Design Techniques

Two novel techniques have been employed in the implementation. The first is to build the functional units (FU) using pass transmission gate (PTG) logic; such logic promises significantly lower power than conventional CMOS whilst delivering high performance. The second unusual technique employed it that the whole system is clock-free. Clockless (or asynchronous) logic is used to eliminate clock generation, buffering and distribution – a major power user – at the system level. Each functional unit is responsible for its own timing; data is passed in and out using handshake signals. This means that a functional unit which is not in use dissipates almost no power. It also allows the implementation of 'unusual' DSP functions – such as Hamming distance calculation or signal clipping – in an energy efficient manner. If evaluation in one unit is slow the whole system will adapt on a cycle-by-cycle basis. The final advantage of asynchronous logic here is that it adapts automatically as the supply voltage is changed; a reduced input voltage gives slower processing but much greater energy efficiency.

## 4. Implementation

CADRE-s is implemented on a 0.18µm CMOS process having 6-metal layers and runs from 1.8V. The tests described seek to demonstrate the energy efficiency of the FU designed by the coherent low energy design techniques described in the previous section. The kernel benchmarks are translated into binary codes by the CADRE assembler. A Verilog module then rearranges this binary code into the format for serially shifting into CADRE-s. The CADRE-s die, shown in Figure2, measures 4.5 x 3.9mm and contains over 230,000 transistors plus 14 RAM memories of 2k x 16 bits plus a further 4 RAM memories of 64 x 32 bits. The results show that the FU designed can achieve an energy improvement by a factor of 5 in the multiply accumulator units and a factor of nearly 2 for the overall system compared with the original CADRE system.

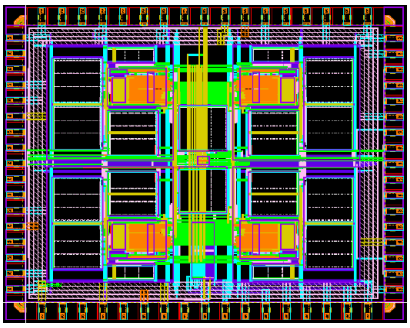

Figure 2 CADRE-s Die Photo

## 5. Conclusion

CARE-successor (CADRE-s) has been implemented as a full custom design and simulations are presented to successfully demonstrate the energy-efficiency of the FU. This demonstrates the importance that energy efficient logic, circuit and layout techniques contribute to a design.

## 6. Acknowledgement

This work was funded by EPSRC grant GR/S61270/01 and the authors are grateful for this support. The authors are also grateful to Dave Clark and Jeff Pepper for encouragement in this work.

## References

[1] M.Lewis and L. Brackenbury, "CADRE: An Asynchronous Embedded DSP for Mobile Phone Applications", *Design Automation for Embedded Systems,* Vol.6, No.4, pp.451-475, 2002.

[2] M. J. G. Lewis, "Low Power Asynchronous Digital Signal Processing", *Doctor of Philosophy Thesis*, Faculty of Science and Engineering, University of Manchester, 2000.

[3] Anantha P. Chandrakasan, Robert W. Brodersen, "Minimizing Power Consumption in CMOS Circuits", Proceedings of the IEEE, vol.83, Apr., pp.498-523, 1995.

[4] D.K.Arvind and Robert D. Mullins, "A Fully Asynchronous Superscalar Architecture", *International Conference on Parallel Architecture and Compilation Techniques,* Oct., p.17-22, 1999.

[5] D. Peiliang, Y. Rilong, X. Hongbo and Y. Chengfang, "Multi-clock driven system: a novel VLSI architecture", *Proceedings 4th International Conference on ASIC,*Oct., pp.555-558, 2001.

[6] M. Singh and M. Theobald, "Generalized latency-insensitive systems for single-clock and multi-clock architectures", *Proceedings on Design, Automation and Test in Europe Conference and Exhibition,* vol.2, Feb., pp.1008-1013, 2004.